The treatment of navigation in web engineering

M.J. Escalona *, J. Torres, M. Mejías, J.J. Gutiérrez, D. Villadiego

Department of Computer Languages and Systems, University of Seville, Seville, Spain

Abstract

This paper aims at giving a global vision of the most popular web engineering approach. Web systems have woken up a high interest in companies and in the research community in the last years. Thus, techniques and methods are being proposed in order to offer a suitable framework to deal with the special characteristics of the web. For these reasons, some years ago a new line in the software engineering appeared. This line, then named web engineering, has grown in the last years, proving that web systems have special characteristics that require a special treatment. One of the most treated characteristic is the navigation. Navigation is a critical aspect in web systems and its suitable development in the life cycle is a basic need to guarantee the project quality. This survey shows how navigation is treated in 15 web proposals, which are the most referenced ones, and it analyses the available techniques, models and the possible gaps in the treatment.

Keywords: Software engineering; Web engineering; Navigation treatment; Tools

1. Introduction

Since the *Net of Nets* was born in the 1970s, as a net to spread research material, an amazing change in the use of Internet has taken place [1]. In the last years, Internet has become a popular tool and the number of users who work every day with it has grown crazily.

Companies and organizations find in Internet a suitable way to present their businesses, and, also, a powerful way to contact with their clients and employees all over the world.

This evolution, the high advance of communications and the increase in the benefits of the equipment, networks and routes of transmission have led to the fact that most of the actual systems are developed or adapted to Internet.

Since the development of software systems in Internet appeared, the research community has detected the necessity of proposing new methodologies, techniques and models to offer a suitable reference environment for the new and special characteristics of Internet. For this aim, a new research line in the software engineering has been developed in the last years: web engineering [2]. Web engineering *is the systematic*, *structured and quantifiable application of methodological proposals to the development*, *evaluation and maintenance of web applications* [2]. At first, the development of web systems was an ad-hoc process. Applications were developed without following any structured process which guaranteed the quality of the results. When the web engineering appeared as a new research line, several new methodological approaches were proposed and some surveys and comparative studies agreed that it was necessary to offer new methodological environments to deal with the special characteristics of the web [3–9].

Nowadays, the research community accepts all over the world the idea that web projects have special characteristics (critical navigation, hypermedia, customization, etc.) that must be dealt carefully in the life cycle and that need their own models and techniques [10,11].

One of the most critical and special characteristic of the web environment is the navigation. The treatment of navigation is an aim which has led several research groups to propose and develop special models to deal with it in

* Corresponding author. Tel.: +34 954 553 867; fax: +34 954 557 139.
E-mail addresses: escalona@lsi.us.es (M.J. Escalona), jtorres@lsi.us.es (J. Torres), risoto@lsi.us.es (M. Mejías), javierj@lsi.us.es (J.J. Gutiérrez), villadiego@lsi.us.es (D. Villadiego).

the life cycle. This paper is focused on the analysis of the actual situation of web engineering in the treatment of navigation. It also shows how some of the most popular web proposals deal with navigation and compares the different approaches.

The main aim is to show that navigation is a critical aspect deeply studied by a high number of research groups, like it was concluded in several comparative studies [3,6], but the paper also tries to show the gaps and the main problems found in these approaches. In this sense, this paper follows the idea of previous surveys that compare web methodologies in order to find their advantages and disadvantages. Comparative studies like [3–7] analyse web proposals from a general point of view. This paper is a survey focused on the navigation treatment.

However, before starting to describe the proposals, it is necessary to outline some important aspects. First of all, the research community does not accept any standard approach. The paper analyses deeply 15 methodologies. Although there are more approaches, these ones are a very representative group. The other ones were not selected in the study for several reasons. Some of them have only been proposed [12] and there are not too much references of them. Other works are mainly focused on the last phases of the life cycle and they do not offer special information for the environment of this work [13–17].

It is also necessary to stick out that the web engineering is a very recent research line. Most of proposals are continually being revised and they are very often extended or corrected. New models, techniques or phases of the life cycle are continually added. The offered study is based on the last information found or even asked to the own authors.

Finally, it is necessary to indicate that one of the main problems found when a comparative study is presented in the web engineering environment, is the variability of the terminology and the lack of standards. However, as it will be described, we can observe a recent eagerness to try to offer some standard aspects.

This study has been divided on two main blocks. In Section 2, the first one presents each studied proposal. For each one, the life cycle is presented with an activity diagram [18] and the treatment of the navigation in the life cycle is outlined. The second block, in Section 3, offers a comparative study and an analysis of situation. Relations and common points in the proposals are analysed in Section 3.1. In Section 3.2, the concept of navigation and its different meanings in each methodology is presented. In Section 3.3, the treatment of the navigation in the life cycle in each proposal is analysed. And, in Section 3.4, techniques proposed by each methodology to deal with navigation in each phase of the life cycle is presented and compared. Finally, in Section 4, the paper details the conclusions and the final analysis.

2. Web engineering proposals

This section presents a short vision of each analysed proposal. In order to present a graphical vision of its life cycle, each proposal description has an associated activity diagram, excepted HDM (hypermedia design model) [19] which has a class diagram.

This short vision is the base for the comparative study in the following section.

2.1. HDM – Hypermedia Design Model

HDM [19] had been the first approach to support the hypermedia information. HDM is not really a development methodology. In fact, HDM is an extension of ERDs (entity–relation diagram) [20,21] to which some new aspects to model hypermedia systems were added. HDM proposes a set of elements that allow the designer to specify an hypermedia application. These elements are *entities*, *components*, *perspectives*, *units* and *links*. All of them can be included in the classical semantic of ERDs. However, although classical elements like entities of ERDs are present in HDM, they had been extended in order to model complex navigational structures with links and internal navigation semantics. In conclusion, a specified application with HDM is a general structure composed by basic units, named entities. An entity represents a physical or conceptual object in the system. Entities are grouped in kinds of entities which are each one defined with a name, a group of perspectives that presents how the information of the entity can be showed and a set of application links to navigate through the information. Entity is the minimal autonomic unit in each HDM model, although there are other concepts. Each entity is composed by hierarchical components which inherit the properties of the entity. Those components cannot exist without their entity. They are the elements that design groups of units and nodes to present the same set of information of the entity.

A unit is a repository of information managed by the application. It represents a fragment of the content of an entity shown in a concrete perspective. In this sense, each perspective lets represent multitude of presentations of the same piece of information (for instance, the presentation of a document in multiple languages). In Fig. 1, a class model shows the relations between these elements.

Nowadays, HDM is not used. Mainly for two reasons: the first one is because the structured paradigm had been

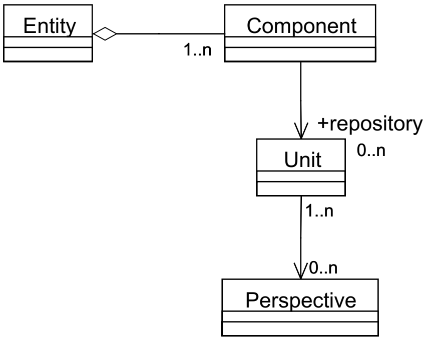


Fig. 1. Relations between HDM elements.

substituted by the object-oriented paradigm. Thus, developers prefer object-oriented methods. Besides, although HDM proposes ideas to develop hypermedia systems, it is not a methodology. It is just a model.

However, although it is not used, HDM has been the base for other approaches. Some methodologies like RMM (relation management method) [22], OOHDM (object-oriented hypermedia design method) [23] and W2000 [24], accepted and adapted HDM ideas in their life cycles.

2.2. RMM – Relationship Management Method

RMM [22], unlike its predecessor HDM, can be considered as a methodology because it covers the complete life cycle. Its process is composed by seven phases that allow to model the structure of the application and its possibilities of navigation.

The proposal is based on the ERD and HDM models, from which RMM defines a new model, named RMDM. It offers a special graphical language to describe domain objects, their inter-relations and the ways to get the hypermedia navigation of the application. Fig. 2 shows the life cycle of RMM.

The process starts with *the development of the E–R model*, without taking navigation or presentation details into account. In this sense, the model is developed as in a non-hypermedia system. Next, the *design of slices* must be done. A slice is a subset of entities attributes which are going to be present together to the user. It will be a view of the system. The slice design is enriched with *the navigation design*. Both make the RMDM model. In base of this model, the process continues with the fourth phase called *defining the protocol of conversion* that allows to go from the RMDM model to the concrete development platform. After that, the next step is *the design of the interface* that is to say the development of screen for the users. Then, we have *the implementation of the application* in the selected computer language, and, finally, the process ends with *the testing of the results*.

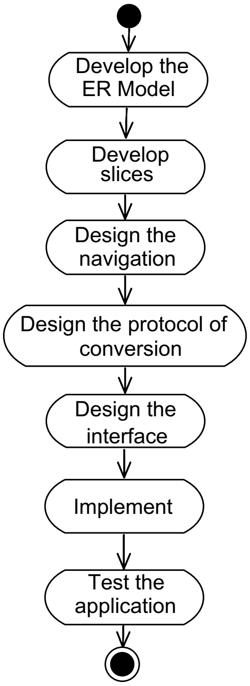


Fig. 2. Development process of RMM.

2.3. EORM – Enhanced Object Relationship Methodology

With OOHDM, EORM [25], in one of the first web proposals, focuses on the object-oriented paradigm. It gets many ideas from HDM, but translates them in the object-oriented paradigm. Its life cycle is a classical one, as shown in Fig. 3.

EORM structures the development process in three phases: *analysis*, *design* and *construction*. The analysis phase does not have exactly the same aims as in other proposals. Analysis phase in EORM is the object design using the OMT model [26]. In this phase, aspects like navigation or interface are not modelled [27].

Then, the design phase modifies the class model obtained previously adding new semantic to model links and navigation. This object model with the navigation aspects is named EORM and it presents the structure of the system and the navigation possibilities.

Finally, in the construction phase, all these aspects are implemented in the concrete computer language.

2.4. OOHDM – Object-Oriented Hypermedia Design Method

Perhaps, OOHDM [23,28] has been the most referenced and studied web methodology in the last years. At the beginning, it was based on HDM but then it has been focused on the object-oriented paradigm. Its main contribution is the high acceptance of its life cycle presented in Fig. 4. The process starts with the *design of conceptual classes*. The static structure of the system is designed at this stage of the class diagram.

The next phase is the *navigation design* which develops a navigational model. The latest offers a view of the

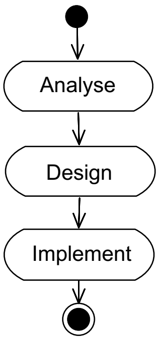


Fig. 3. Development process of EORM.

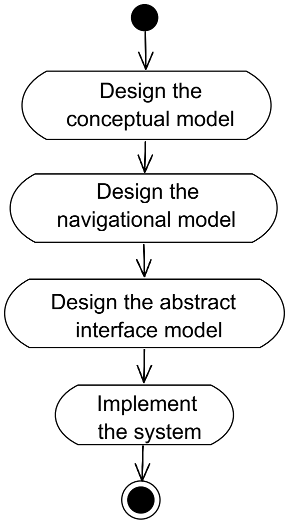

Design the conceptual model
Design the navigational model
Design the abstract interface model
Implement the system

Fig. 4. Development process of OOHDM.

conceptual model that expresses how the user can navigate through the information modelled in the first stage.

In the third phase, the *abstract interface design*, a third model, named abstract interface model, is designed. This model describes how the information will be presented to the user. Thus, the navigation model is a view of the conceptual model and the abstract interface model is a view of the navigational model.

In the last phase, the *implementation* of the three models is programmed.

In each above mentioned phases, OODHM offers specific models languages to represent these models. Some of them, like the navigation context or the abstract data views are not used nowadays. However, the research community widely accepts other ideas of OOHDM, like representing navigation model with a special class diagram named navigational class diagram.

Separating the modelling of different aspects in hypermedia systems is another very accepted idea of OOHDM. Most of proposals defined before OOHDM widely accepted and assumed the idea of modelling conceptual, navigation and abstract interface aspects in a separate way.

Finally, it is necessary to stick out that OOHDM is not a static proposal [29]. Nowadays, it is being improved and enriched. For instance, it has been extended with a requirements phase using an own technique named UIDs (user interation diagrams) [30–33].

2.5. WSDM – Web Site Design Method

The method of designing web sites (WSDM) [34] is a proposal that defines the system according to the users groups. Its life cycle is divided into four phases: *user's model*, *conceptual design*, *implementation design* and *implementation*. As well, the user's model is divided into two subphases: *classification* and *description* and the conceptual design into two phases: *object modelling* and *navigational design*. WSDM life cycle is presented in Fig. 5.

The development of the user's model studies possible roles of users who are going to interact with the system and also their relations. In the next phase, and according to the previous users' classification, the conceptual model of the system is developed. However, the conceptual model does not have the same meaning as in OOHDM. During the conceptual modelling two tasks are applied together; the object modelling, that is the same as the conceptual modelling in OOHDM, and the navigation modelling, that is the same as the navigational modelling in OOHDM. Besides, in WSDM, more than one navigational model exists. Each role detected in the first phase has its own model.

Then, the implementation design phase models the specific interface for each role. Finally, in the implementation phase all these aspects are codified in the selected language.

WSDM is also a live methodology. It is being changed and adapted to new requirements. Nowadays, one of the most interesting works of this research group is the development of a CASE tool that lets apply the WSDM life cycle [35,36].

2.6. SOHDM – Scenario-based Object-Oriented Hypermedia Design Methodology

SOHDM [37,38] is one of the first web proposals that deals with requirements. Its main contribution is its life cycle because it starts with the application of scenarios like a technique to elicit and define requirements [39].

Its development process is divided into six phases presented in Fig. 6.

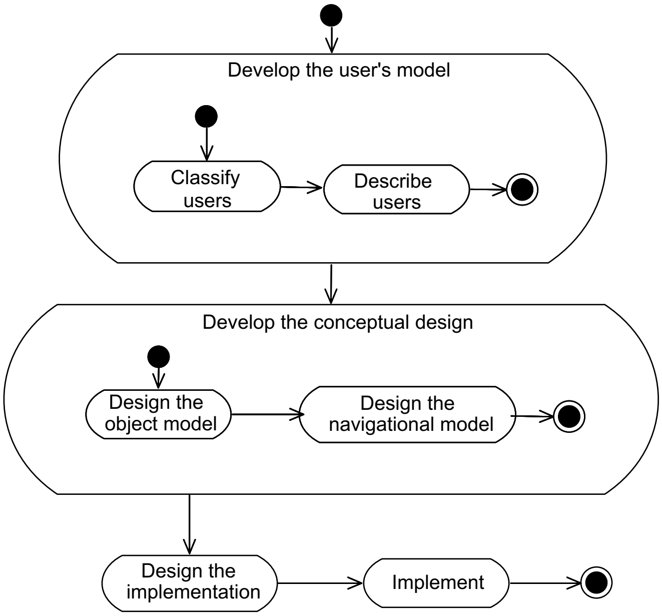


Fig. 5. Development process of WSDM.

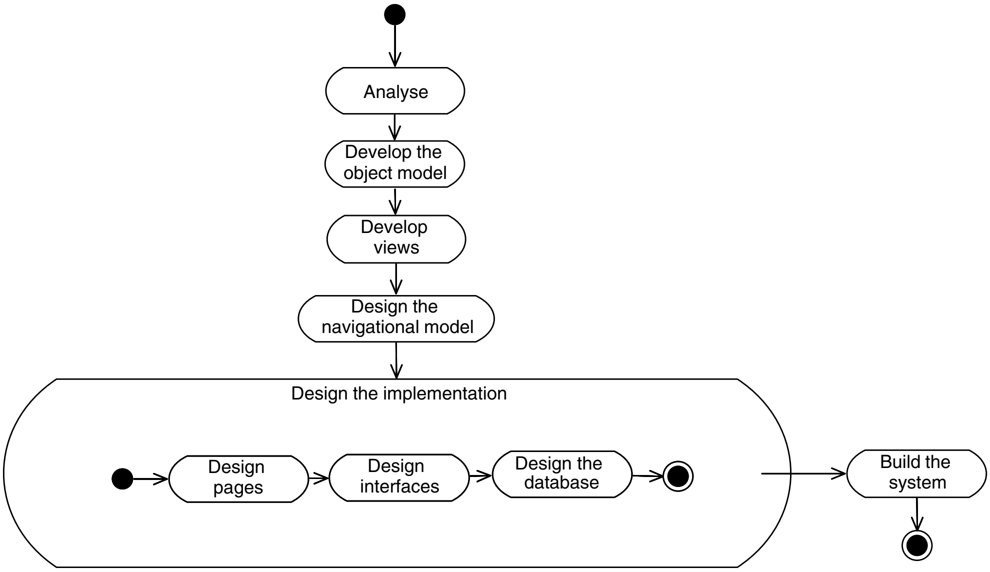


Fig. 6. Development process of SOHDM.

The process starts with the *analysis phase* in which a study about the necessities, the environment and actors must be done. The main objective of this phase is to design scenarios that describe activities in the system. Then, the phase "*object model*" develops a class diagram to describe conceptual information. In the next phase, *the development of views*, objects are reorganized in navigational units that represent a view of the system objects. In the *design of the navigational model*, these views are enriched with links and hyperlinks. After that, web pages, interface and the database are designed, in the *implementation design phase*. Finally, the *construction phase* builds the application.

2.7. RNA – Relationship-navigational analysis

Contrary to other proposals, RNA [40] does not cover several life cycle phases. Its development process is focused on the analysis phase. However, in this approach, the concept of analysis is not exactly the same as in other proposals. In fact, this approach includes activities in the analysis while other approaches include its in phases like the requirements or the design ones.

Fig. 7 shows a diagram with its life cycle. It starts with an *environment analysis*, in which the audience characteristics are studied and classified and the different roles of the system are analysed. After that, *interest elements* like screens, documents, etc. which are going to be required during the development process, are defined. The process continues with the *knowledge analysis*, which develops a scheme to represent the system. In the latest, objects, processes and methods that have to be offered are identified. Then, an *analysis of the navigation* describes how the user will be able to navigate through the system. And to finish, the *analysis has to be implemented*.

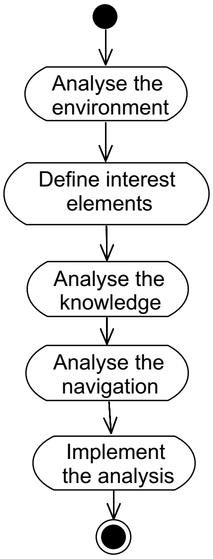


Fig. 7. RNA development process.

2.8. HFPM – Hypermedia Flexible Process Modelling Strategy

HFPM [41] is one of the most complete proposal because it covers a deep life cycle. In fact, it does not offer new models or techniques and assumes many OOHDM's ideas. However, it covers a complete development process, and even generates documents. In Fig. 8, its life cycle scheme is presented. In this process, each phase is divided into tasks and subtask, offering a very detailed development guide.

The process starts with the development of a *requirements model* and a *planning* and continues with the phase of *conceptual modelling*, *navigational modelling* and *abstract*

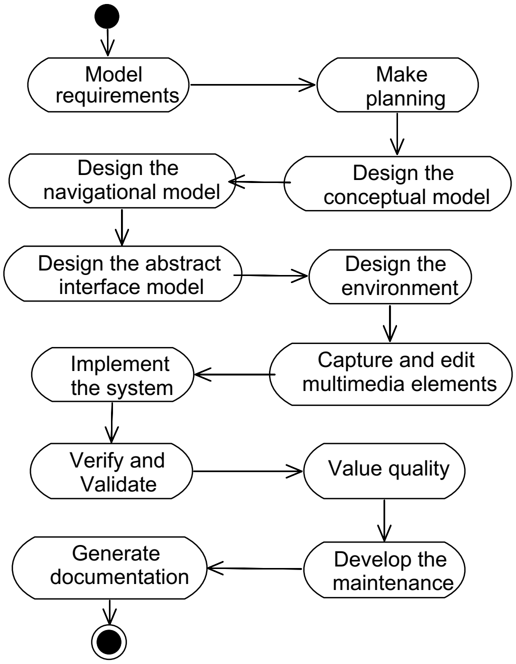


Fig. 8. Development process of HFPM.

interface modelling, which are similar to the OOHDM ones. The next step is the *environment design* and the *capture and edition of the multimedia elements*, with the description of the multiple media that appear in the system and also storage systems which will be used. Thanks to that knowledge, the system is *implemented*. The *validation* and the *verification* are then applied like some metrics to *measure the quality* of the system. Finally, the process ends with the *maintenance* and *documentation generation* phases.

2.9. Building web applications with UML

The Conallen's proposal [42] is a suggestion which has gotten a good acceptance in the last years. It is completely based on UML and on the unified process in its life cycle. It defines a high number of its own UML extensions and specific stereotypes for the web.

Its life cycle starts with a previous *phase of planning*. Then, a cyclic process covering *requirements treatment*, *analysis*, *design*, *implementation*, *test* and *evaluation* phases starts. When the system is accepted, the phase of maintenance starts. This process is shown in Fig. 9.

Basically, the models and techniques offered by Conallen are inherited from UML. However, they enrich and define several new specific model elements for web environment like the possibility of representing java beans, frames, bottoms, etc. These new stereotypes can be included in rational rose and used easily during the modelling.

2.10. WebML – Web Modelling Language

As it is defined by its authors [43,44], WebML is a notation to specify complex web sites in the conceptual environment [45]. It gives a description of the web from different points of views: conceptual, navigational, presentational, etc.

Its life cycle is presented in Fig. 10. The development process starts with the *conceptual modelling* of the system. In this phase, in which using a model language like UML, WebML does not specify any concrete, the static structure of the system is presented. After that, we deal with the development of an *hypertext model* in which hypertexts that can be published on the web are described. Each hypertext defines a view of the web site. The description of hypertexts is made using two models: the composition model, that defines the pages in the system, and the navigation model, that describes the navigation through them. In the next step, the presentation model, which defines the physic appearance of the web pages, is developed. And, finally, the personalization model underlines how the system has to be adapted to each role of user.

In order to conclude, we have to stick out that one of the most interesting contribution of WebML is that it offers a tool CASE named WebRatio that lets apply the proposed techniques and get the results systematically [46].

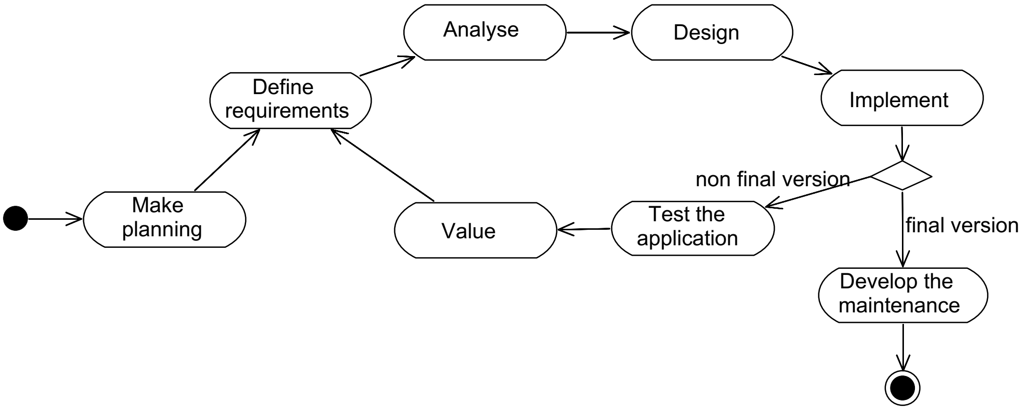


Fig. 9. Conallen's development process.

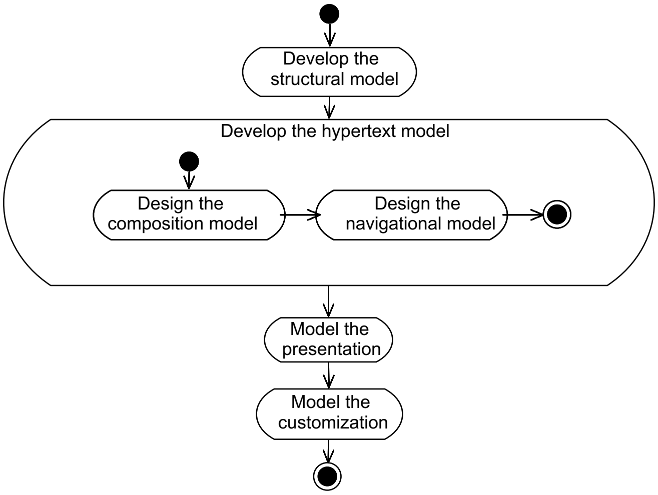

Develop the structural model
Develop the hypertext model
Design the composition model
Design the navigational model
Model the presentation
Model the customization

Fig. 10. WebML's development process.

2.11. UWE – UML-Based Web Engineering

UWE [47] is a methodological proposal based on the unified process [48] and on the UML to develop web applications. UWE covers the complete life cycle of web applications and focuses its work on customized and adaptive systems. Its development process is divided into three main phases: the *requirements capture*, the *analysis and design* phase and the *implementation* phase.

The development process of UWE is mainly focused on the analysis and design phase. This development process is completely detailed and defined, also the structure of these models. A scheme with its general phases is presented in Fig. 11.

UWE makes a very detailed description of its phases. Thus, although it was not presented in the figure in order to avoid making it too much complex, UWE proposes the next phases:

1. In the requirements phases, UWE proposes to start with the *users' identification* and the *elicitation of requirements*. It deals with requirements depending on if they are information requirements, navigation requirements, adaptation requirements or user interface requirements, and also some related additional requirements, like hardware or security constraints. Then, it focuses the work on the *development of use cases*, the *generation of glossaries* and the *development of prototypes* of the user interface.
2. The analysis and design phase is quite similar to the OOHDM one. However, UWE covers more aspects than OOHDM. In fact, UWE separates *conceptual design*, *user model*, *navigation design*, *presentation design*, *adaptation design*, *design of the architecture* and the *detail design* of classes, subsystems and interfaces.
3. Finally, in the implementation phase, UWE includes tasks that let the implementation of all the previous models: *architecture implementation*, *hyperspace structure implementation*, *model user implementation*, *adaptive model implementation* and all the tasks that deal with the *integration* of these implementations.

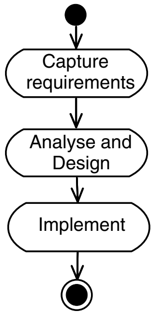


Fig. 11. Development process of UWE.

Focusing UWE's study on the treatment of navigation, UWE proposes in its requirements phase an elicitation of navigation requirements. UWE considers navigation requirements like functional requirements and, although it does not propose specific techniques for its treatment, the navigation requirements treatment is mainly made using use cases. UWE separates the navigational requirements in order to identify better relevant aspects for the navigational model in the analysis and design phases.

This navigational model is built in two steps. In the first one, a *navigation space model*, built like a view of the conceptual model, is developed. This model shows that classes and models are accessible using a special class diagram with special classes named navigational classes. These are UML classes stereotyped with special tags.

This model is enriched in the second step with the *model of the navigation structure* which presents the way to visit the accessible elements. The final navigation model includes both submodels and using then, it details the accessible elements and the way to navigate between them.

In the last years, UWE proposal has getting a big acceptance in the research forums. Its models are completely based on UML, so they are being very good valued. Besides, UWE is an alive proposal. Nowadays, they are working on a tool to support its life cycle, named ArgoUWE [49].

2.12. W2000

The W2000 proposal [22] has been the evolution of the object-oriented paradigm of HDM. W2000 is different to HDM in two basic ideas. The first one is that HDM was only an enriched model of ERD, but not a methodological proposal. W2000 proposes a life cycle to develop web systems. The other big difference is that W2000 is focused on the object-oriented paradigm. However, despite of these differences, HDM assertions are been accepted in W2000 and adapted to the object-oriented paradigm.

The W2000 life cycle is shown in Fig. 12. The process starts with a *requirements analysis phase* mainly based on

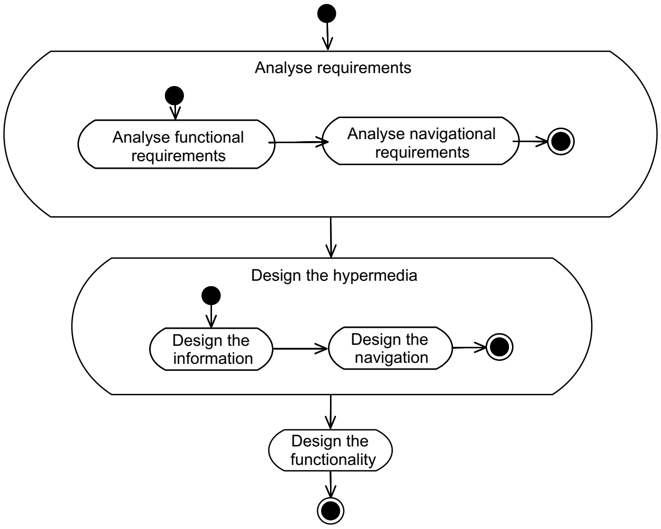


Fig. 12. Development process of W2000.

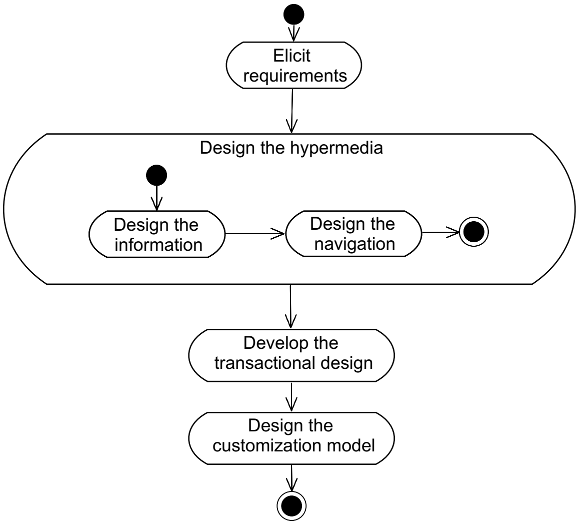


Fig. 13. Development process of UWA.

use cases. Thanks to the knowledge of this requirements phase, the process continues with the *hypermedia design* phase. In this phase two models are developed: the *conceptual* and the *navigational* model. For that, W2000 modifies and extends some UML models like the class diagram and the state diagram. The last phase is the functional design phase, in which the sequence diagram is used to express the functionality of the system.

An important characteristic of W2000, which is outlined in the next section, is the separation of the navigation aspect and the information structure since the very firsts phases of the life cycle. As it is shown in Fig. 12, W2000 proposes a functional and a navigational requirements analysis. However, the used technique in both kinds of analysis is the use cases one, without specifying how both kinds of requirements can be separated, dealt, identified or elicited.

2.13. UWA project – Ubiquitous Web Applications

The UWA project [50] is a European project in which several research groups worked together several years ago. In fact, the most of approaches, techniques or models presented in its life cycle is an heritage of other approach like W2000.

The UWA development process is divided into four big phases, as it is shown in Fig. 13.

1. *The elicitation of requirements* is the phase that defines the functional requirements. It is a requirements elicitation based on objectives. The process starts defining the high level requirements and, from them, the necessities of the systems are systematically concreted.
2. *The hypermedia design*, is the phase in which data, navigation, data presentation and users' services are modelled. This phase comes from W2000. In fact, UWA proposes three models to represent the hypermedia: the information model, the navigation model and the presentation model.
3. The *transaction design models*, the behaviour of the interaction transactions and their effects in the system. Transactions are modelled according to two point of views: the static point of view, which studies what transactions can appear, and the dynamic point of view, which analyses the effects of these transactions.
4. The customized design analyses how the application has to be adapted to the context in which it is executed. This phase is an heritage of WUML [16]. This web proposal is not included in this study because it is only focused on the last phases of the life cycle. It uses components like implementation technique. UWA, in fact, proposes to design the customization with components.

2.14. OOH – Object-Oriented Hypermedia Method

OOH has got as main objective, the *application of the object-oriented paradigm to capture the semantic to the efficient modelled of user interfaces and its implementation on the web environment* [51]. The idea of OOH is to offer a methodological environment to let the systematic model of adaptive interfaces being suitable for the group of users.

Its life cycle, as shown in Fig. 14, starts with a *requirements analysis* and is followed by an *engineering* phase. In the latest, the analysis and the domain and navigation design, and also the presentation design are developed. Then, there is a *construction and adaptation* phase in which the final system is obtained with a set of templates. The last stage includes the interface *evaluation* by the user. All these phases are very focused on the user's roles.

A very important characteristic of OOH is that it designs its interfaces and its navigation aspects adapting its to each user's role and his/her specific necessities.

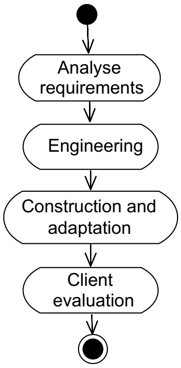


Fig. 14. Development process of OOH.

But, perhaps one of the most interesting aspect of OOH is its tool, named Visual WADE [52], that covers the complete life cycle of OOH.

2.15. NDT – Navigation Development Techniques

NDT is a very recent proposal [53–55] based on some previous comparatives studies. NDT is a proposal that only works in the requirements and analysis phases. It offers specific methods and techniques to deal with web requirements and to get from them the analysed models applying some systematic process.

The life cycle of NDT is presented in Fig. 15. In the first phase, the *requirements phase*, each kind of requirements is treated depending on its kinds. NDT supports storage information, actors', functional, interaction and non-functional requirements. For each of them, NDT proposes a specific pattern, that is to say a structured template which defines specific and necessary information that describe a kind of requirements. Patterns are the base of NDT. Patterns give a very structured definition of very useful requirements to propose systematic processes that allow to derive analysis models from the requirements patterns.

In the *analysis* phase, NDT propose to develop three models: the conceptual model, the navigation model and the abstract interface model. In this sense, a strong influence by OOHDM can be detected in NDT. However, the biggest influence on NDT is UWE. NDT assumes UWE notations as models. In this sense, NDT proposes a new phase of requirements but, after that, it is possible to complete the rest of the process using another web proposal, mainly UWE's one. That is because NDT assumes that posterior phases of the life cycle have been being treated enough in the web engineering and tries to give support in the less treated phase, the requirements one.

The systematic generation of analysis models has actually two phases. In the first one, analysis models are systematically generated from the requirements models, these are the basic analysis models. After that, analysts can make some changes to these basic models in order to adapt them better to the reality of the system. However, NDT manages the possible changes in order to check that there are not inconsistencies between the final models and the requirements.

Lastly, two important points about NDT have to be stuck out. The first one is that NDT has a tool, named NDT-Tool [56], that allows to apply all the NDT techniques, to get final results and to apply automatically generation processes. This tool is compatible with rational rose and generates models and results automatically.

The second point is that NDT is a very practical methodology. It has been applied in several real projects in the last years, getting very good results [57,58].

2.16. Other studied approaches

In this survey, we have presented a short vision of fourteen proposals. However, this sample is only a subset of the high number of web engineering proposals that have been presented in the last years. This survey only gives the closest proposals to the treatment of navigation or the most

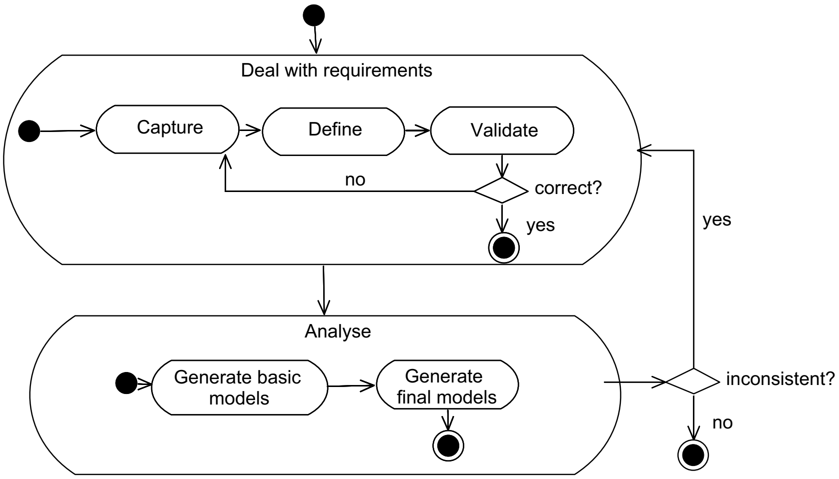


Fig. 15. Development process of NDT.

original ones in order to study the advance of the navigation treatment in web engineering. However, other web proposals were studied during this work but finally they were not included because they are too recent and their life cycle is too focused on the last phases of the life cycle or because there are not too much available information of them.

Although the selected group of proposal is very representative, in any surveys there are always interested proposals which are not included. That is why, to finish this short presentation, some more studied proposals are named and shortly described in this section, although they are not included in the next comparative study.

1. *OO-method and OOWS (object-oriented web solutions)*: The OO-method proposal was initially developed for software projects in general. But, in the last years, it has been being adapted for the web environment. One of its more interesting adaptations was named OOWS [12]. OOWS is a proposal for the treatment of requirements and navigation being applied in web systems. However, although this proposal is very closed to this survey, it is too recent yet.
2. *Araneus project*: The approach proposed by Paolo Atzeni and Paolo Merialdo from the University of Rome and Giasalvatores Mecca from the University of the Basilicata [59], named Araneus project, is based on a model named Araneus. This is a specific data model to describe schemes of applications with hypertexts. It is highly oriented to the database and E/R models environment.
3. *The MacWeb hypermedia design method*: The hypermedia development environment MacWeb, proposed in 1995 [60], cannot be exactly consider as a methodological proposal. In the MacWeb environment, an hypermedia application is developed according to the interface. Its authors stick out that the most important aspect of hypermedia systems is the communication with users. Thus they focus the development process on the development of the interface. This interest for the communication is perhaps one of the most important aspect of this proposal because, as it is going to be concluded in the next sections, this aspect is being considered more important every day.
4. *OO/pattern approach*: This approach [61] is quite similar to HFPM because both of them propose the use of patterns and the object-oriented paradigm to the navigational and the interface design. However, this proposal does not cover the complete life cycle unlike HFPM. It is interesting because is the first approach that proposes the use of use cases to analyse the application.
5. *OSM (object-oriented system model)*: Actually, it is not a methodological proposal. It is an object-oriented model that tries to be enough solid to be applied in all life cycle phases: specification, analysis, design, implementation and evolution [62]. It proposes to present web system with three models: the object model, the behaviour model and the presentation model [63]. It is a very original proposal because it is quite different to the rest, however it is not very referenced.
6. *Design-driven requirements elicitation*: This proposal, mainly focused on requirements, is part of the process proposed by Lowe and Eklund [64,65] to develop web applications. It proposes to divide the treatment of requirements into capture, definition and validation of requirements. One of the most important aspect of this approach is that it was proposed according to the results obtained applying it in real projects.
7. *WUML (web UML)*: It is also a proposal that is very close to the last phases of the life cycle [17]. In fact, other general proposals include a part of WUML, like the UWA project, in order to assume the more concrete phases of the life cycle. It is based also in the use of UML like the graphical language to mode.
8. *WebComposition*: It is a very particular proposal because it is based on the use of components [15]. Its life cycle, totally focused on a design phase that is very close to the implementation one, starts with the development of components that are then divided and prototyped several times until the implementation is gotten.

3. Comparative study

In this short abstract, a general vision of the life cycle of the most representative web proposals has been presented. As it has been demonstrated by several surveys [3–5,53,66,7,51,67], these proposals have several common points to make comparative studies. However, this survey is focused on the comparative study of the treatment of the navigation. In this section, some different aspects in the treatment of the navigation are going to be analysed.

3.1. Relations between methodologies

In the presented methodologies, we can detect some common points in the process of developing web systems. It is because some proposals are based on other previous ones and they assume previous ideas, techniques or models. In this section, the relations between the studied web proposals are going to be analysed in order to stick out differences and similarities and to know which has been and is being the chronology evolution on the web methodologies.

In the previous sections, methodologies have been presented in a chronological way. A global schema of their history is presented in Fig. 16. Nowadays, some of these approaches, concretely HDM or RMM, are not used. They are the only ones that are not object-oriented, for this reason they appear shaded.

Fig. 16 also presents relations between the proposals. As it has been said, although every approach is different and their idea or conception of the problem is particular, some proposals are based on previous experiences. This relation is shown in the figure with continuous directed lines. Thus,

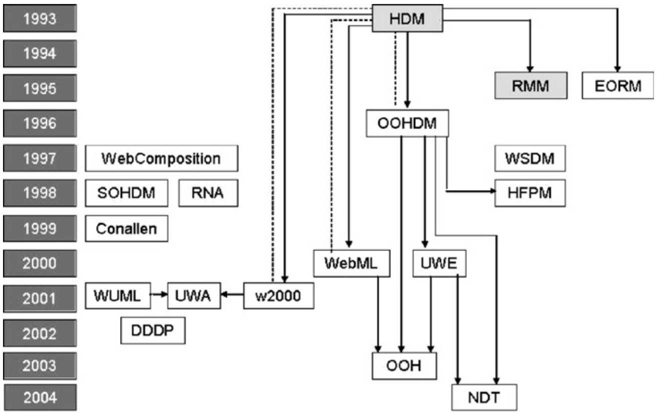


Fig. 16. Relations between proposals.

the figure indicates that UWE has been influenced by OOHDM, like it was introduced in the presentation of UWE in Section 2.11.

In other cases, the relation between methodological environments is stronger because the same research team, or part of it, has worked in several proposals. There are even some approaches, like the UWA project, that assumes completely a web proposal, like W2000 in the UWA case. In the figure, this kind of relation between approaches is presented with a dotted line.

In the figure, we can also observe that the publication of HDM was a landmark in the web engineering environment. Nowadays, HDM is not used. However, the necessity that raised by offering new ways to work in the hypermedia environment, presenting new model elements, was an inspiration for other research teams. The questions suggested by HDM nourished other very important methodologies like OOHDM, W2000 or WebML, which, have also been used as a source of inspiration for later proposals.

Another important landmark was OOHDM. This, with HDM, is, perhaps, the most referenced proposal. The idea of concepts separation and also the use of navigational class diagrams are the most accepted proposals by other research groups. In fact, UWE, OOH, HFPM or NDT assumed them in their life cycles.

3.2. The concept of navigation in web engineering

In order to study how the navigation is dealt in the web environment, a concreted definition of navigation has to be presented. Actually, there is not a standard definition of navigation and it is difficult to make a good definition for all the proposals.

As it can be observed in Fig. 16, HDM is one of the most referenced approach. It was the first proposal in sticking out that the navigation is a critical aspect in hypermedia systems that must be dealt with care. HDM defined navigation like a view of entities, it is represented by a set of components that shows a view of the static structure, that is the ER model.

The idea of studying the navigation like a view of the static structure of the system, and also the fact of assuming it like a critical aspect in hypermedia system, has been assumed by all the posterior web approaches. However, the concept of navigation is not the same for all of them.

It is very difficult to define the concept of navigation in a suitable way for all the proposals. OOHDM, for instance, defines the navigation as a view of the conceptual structure of the system, but this vision is not only a static one. OOHDM says that the structure of the navigation also depends on the context of navigation in each moment. This idea is also followed by UWE. UWE defines navigation as the aspect that shows which objects can be visited by the user and how he/she accedes to them.

Other approaches, like WSDM, define navigation as the possibility of using the information in the conceptual model according to each user's necessities. WSDM defines the navigation according to the users and studies a navigational system for each one.

SOHDM, as it has been presented, defines the necessities of the system on the base of scenarios. Thus, for SOHDM, navigation is the way the user has to follow in order to use the scenarios information.

Another interesting vision, closer to the design, is the definition of WebML. The latest says that the navigation presents as many pages are linked to the conceptual model contents that they show.

But, perhaps OOH definition is the most suitable. This approach defines two kinds of navigations: the semantic and the structural one [68,69]. The first one represents a change in the election of the user, a user activation of a link that accesses to another functional requirements. The structural navigation, closer to the WebML definition, is the voluntary action of a user that causes a change in the interface.

At last, to the question of *what is navigation*, the answer is that it depends of each methodology. This question was presented in different research forums and there was not any consensus. It is clear that navigation in an application is the possibility of going from one point of the system to another one *navigating* thought hyperlinks. However, what is navigation in the modelling of a navigational system is a question without a standard answer for the research community.

3.3. Navigation in the life cycle

In the short abstract of the previous section, the life cycle of each proposal has been presented. As can be observed, there are multiple and different types of life cycles but in each of them, the navigation is dealt in one way or another. We have to compare them in each proposal to study in which moment of the life cycle the navigation is considered to stick out the tendency.

For that, a global life cycle has to be selected as a guide to compare the other ones. The selected one is the unified process [70] because it is a standard approach. It presents a life cycle composed by a set of phases. In each phase, there are some iterations in which a group of workflows has to be applied. Those workflows are requirements, analysis, design, implementation and test. Each of them is more or less important depending of the phase. For instance, the workflow of requirements is more important in the first phases that in the last ones.

Each workflow has a set of objectives and a set of task that must be followed to get the objectives. In base of the unified process approach, workflows that are going to be assumed in the next study are defined in Table 1.

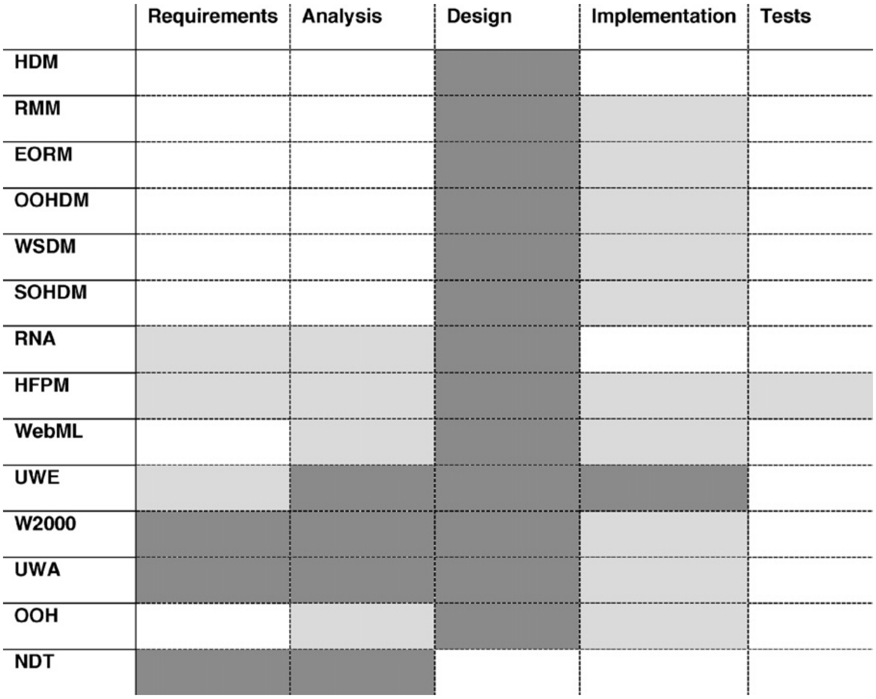
With this definition and the presentation of the life cycle of each proposal in the previous section, Table 2 has been set up. In rows the web proposals are numerated and in columns the workflows of the unified process are presented. The completely shaded cells indicate that the methodology deeply deals with navigation in this phase. That is to say, the approach proposes special techniques, phases or models to treat it. Middle shaded cells express that the methodology deals with navigation in this phase but not deeply. In this case, the study proposes to deal with navigation but does not offer any specific techniques to work with it. White cells means that the proposal does not contemplate the treatment of navigation in this workflow (Table 3).

The idea of using colours allows to observe a chronological evolution in the treatment of navigation. The navigation has been mainly worked in the design phase. The most of the

Table 1
Workflow definition by RUP

Requirements engineering	Definition	It covers the process of finding out what must be built
	Aims	To guide the development process with a suitable description of requirements
Analysis	Definition	It analyses, refines and organizes them
	Aims	To get a more concrete knowledge of requirements and their definitions in order to get a better understanding in the next phases
Design	Definition	It models the system, including its architecture, in order to cover defined requirements
	Aims	To get a depth understanding of the aspects related with requirements and their constraints, and, also, to create a suitable start point for the implementation
Implementation	Definition	It implements in a concrete computer languages the design results
	Aims	To develop the architecture defined in the design
Tests	Definition	It verifies the implementation results testing each construction
	Objetives	To design, implement and execute tests analysing the results

Table 2
The treatment of navigation in the life cycle



	Requirements	Analysis	Design	Implementation	Tests
HDM			■		
RMM			■	▨	
EORM			■	▨	
OOHDM			■	▨	
WSDM			■	▨	
SOHDM			■	▨	
RNA	▨	▨	■		
HFPM	▨	▨	■	▨	▨
WebML		▨	■	▨	
UWE	▨	■	■	■	
W2000	■	■	■	▨	
UWA	■	■	■	▨	
OOH		▨	■	▨	
NDT	■	■			

Table 3
The most used techniques

	Use cases	Scenarios	ER enrichment	Navigational classes	Navigational contexts	Link matrix and ASN model	Design patterns	Textual descriptions based on patterns	Prototypes
HDM			✓						
RMM			✓						
EORM				✓					
OOHDM	✓			✓	✓		✓		
WSDM				✓					
SOHDM		✓				✓			
RNA								✓	
HFPM	✓			✓			✓	✓	✓
Conallen	✓			✓					
WebML				✓				✓	✓
UWE	✓	✓		✓			✓	✓	
W2000	✓			✓					
UWA	✓			✓					
OOH				✓					✓
NDT	✓			✓				✓	✓

analysed proposals include it. However, it is necessary to indicate that this design is, in fact, very closed to the analysis phase. Thus, the conclusion is that the most worked phases in web engineering is analysis and design ones.

Besides, selected proposals are mainly focused on the first phase of the life cycle. For this reason, the implementation and test phases are not so important. In the research community, there are other proposals, like WebComposition or WUML that are more focused on these phases. If we add them to this table, the analysis would change to conclude that there is also an important work in implementation. However, they were not included because they do not offer specific techniques for the navigation.

In any case, studying the table, the main conclusion is that navigation has not been dealt too much in the first phases of the life cycle. This tendency has been changing in the last years, as can be observed. Everyday, research groups in web engineering are being more interested in enriching their proposals with specific techniques or models for navigation at the beginning of the project.

This tendency is waking up new questions in the treatment of navigation in the first phases of the life cycle. The first one is if design is the best phase to start dealing with navigation. Some groups are not agreed with this classical tendency. For instance, W2000, NDT or the UWA project begins the treatment of navigation in the requirements phases. They offer specific requirements techniques to deal with it. Other approaches, like UWE, propose to deal with navigation like a special kind of functional requirements using use cases. Others, like RMM, do not detect navigation requirements like a special kind of requirements.

Another important question is the relation between navigation and roles of users. In the most common idea, since the beginning of the web engineering, navigation is simply a view of the conceptual model. This idea was proposed by OOHDM. However, some actual proposals, like OOH or reviewed versions of old approaches, like the last versions of WSDM, propose to *customize* the navigation to each role. Thus, the navigational model is a group of navigational model, one for each specific role.

Finally, everyday web systems are being more complex, thus their navigational systems are being everyday more complex. It is more necessary to include the expert, who better knows the constraints and the necessities, in the navigation modelling. This expert is not the development team, in fact he/she is usually the final user who is not expert in computer science or in software modelling. For this reason, to model the navigation, some research groups are opening their work environment and are enriching their proposals in order to offer suitable models and techniques to make easier the communication with the user.

3.4. Techniques to deal with navigation

The lack of consensus in the treatment of navigation is also observed in the techniques that deal with it. According to Fig. 16, one of the most influential proposal was OOHDM that proposed to define navigation using two models. The first one, named navigational model, is a view of the conceptual model presented by a special class diagram. The second one, named context diagram, presents how information change in the navigation depending on the context.

Modelling the navigation with a special class diagram was one of the most accepted ideas in the web engineering. OOHDM continues with this idea, and also other proposals like UWE, OOH, NDT, WSDM or HFPM. However, classes in their diagrams are different in all of them.

Use cases are the most accepted technique that deals with requirements. They are used in their standard definition, like in UWE or OOH, or even with specific refining in order to adapt them to the navigation, like in W2000 approach.

In other phases, like analysis or design, defining navigation using a class diagram with navigational classes is the most applied idea. However, this navigational class diagram or even the navigational classes are not the same in all approaches. The actual tendency is to use representations based on Standard notation, like UWE or Conallen's proposal, which use stereotypes of UML to define special classes to represent navigational concepts as links or nodes.

Closer to the design or implementation, perhaps the most important technique which has been evolving in last years is the design patterns. Although, again, there is not a consensus and each proposal proposes its own design patterns.

In order to validate the navigation, the use of prototypes or reviews based on textual descriptions is the most used technique.

From that, we can conclude that although there are some consensus, like the use of navigational classes or design patterns, a high number of techniques and models are specific to one or another methodology, for instance the navigational contexts of OOHDM or the link matrix of SOHDM.

4. Conclusions

As conclusions to this survey about the treatment of navigation in web methodologies, we can stick out that the web engineering is a very young field with a relative lack of ripeness. There is a high ambiguity in the definition of the life cycle in the activities or tasks that must be covered, in the models or techniques that must be applied or in the aspects that must be treated [51].

This general ambiguity is also observed in the treatment of the navigation. Navigation is presented as a very critical characteristic in all the hypermedia and web methodologies, which deals with it. All approaches stick out the importance of the navigation in the web environment and the necessity of a suitable treatment of it. However, some ideas has to be polished up yet.

An important point to be considered is that the navigational work has been mainly focused on the analysis and design phases. An empirical experience proves that, everyday it is being more important to capture navigational requirements because of the high increase of the web systems. In this sense, some question can be solved as a conclusion.

(i) *Must requirements related with the navigation be dealt in the requirements engineering*?
From the empirical results, we can deduce an affirmative answer. Navigational systems are being more complex everyday and in order to guarantee the quality of the system, all the requirements have to be treated correctly since the beginning of the project.

(ii) *If the treatment of the navigation is moved to the first phases of the life cycle, must specific and new techniques dealing with the navigation be proposed*?
According to the previous analysis, some approaches propose to use classical techniques, like the use case, to deal with navigation. However, according to the most recent proposals and research works, use cases or other similar techniques are not enough and they are to ambiguous to treat the navigation.

(iii) *Is necessary to include the final user or the client in the development of the navigation*?
The empirical experience proves that the answer is yes. Web systems, and obviously also their internal navigational systems, are being more complex everyday. Final users or clients are those who really know the navigational necessities of the system. For this reason, techniques that deals with navigation have to be detailed enough to be useful for the development team. But, they also have to offer agile and easy validation systems for final users.

Another important point is the necessity of standards. From this survey and from some similar comparative works [3–5,7,53,51,67], we can deduce that there are too many methods, techniques or terminology to define a same aspect. Research groups must work together in order to find a common definition for the navigation concepts. Used models, techniques or even the life cycle, must be common in a standard proposal. Future works of web engineering have to try to solve really non-treated and necessary aspects and to assume accepted points.

Finally, this work cannot conclude without sticking out a very important point. The development of navigational systems is oriented to the web systems. Web systems have very often to be adapted, changed and updated and they are critical about the high and heterogeneous number of users that deals with. For this reason, it is necessary to offer methodological environments to support the maintenance of web system in the development process.

In most of the presented web approaches, there is a lack in the use of tools that give support to the application of techniques, in the generation of results and also in the validation of models and final results. It is very important and necessary to offer systematically process or even automatic ones to define, generate or even validate results. In this sense, the most recent works of some of these groups are oriented to the model driven architecture. The possibility of changing systematically from independent platform models to dependent platform models is a new research line for the web engineering that in the last months, has been opening new areas in the web engineering, although there are not many works yet.

References

[1] Cazorla A, Carrasco J. Evolución de internet. Comunicaciones de Telefónica I+D, 2001.

[2] Deshpande Y, Marugesan S, Ginige A, Hanse S, Schawabe D, Gaedke M, et al. Web engineering. J Web Eng 2002;1(1):3–17.

[3] Koch N. A comparative study of methods for hypermedia development. Technical Report 9905, Ludwig-Maximilian-University, Munich, Germany.

[4] Retschitzegger W., Schwinger W. Towards modelling of data web applications – a requirements perspective. In: American conference on information systems AMCIS 2000, USA, vol. 1, 2000. p. 149–55.

[5] Barry C, Lang M. A survey of multimedia and web development techniques and methodology usage. IEEE Multimedia 2001(April–July):52–6.

[6] Escalona MJ, Torres J, Mejías M. Metodologías de desarrollo de sistemas de información en la web y análisis comparativo, Novática. Revista De la Asociación de Técnicos de Informática, 159, 2002. p. 49–59.

[7] Lang M. Hypermedia system development. Do we really need new methods? In: Site-where parallels intersect. Informing Science, 2002. p. 883–91.

[8] Gu A. Extending object-oriented modelling languages for web applications. MSC Thesis, University of Technology, Sydney, 2001.

[9] Gu A, Henderson-Sellers B, Lowe D. Web modelling languages: the gap between requirements and current exemplars. In: 8th Australian world wide web conference, 2002.

[10] Dustin E, Rashka J, McDiarmid D. Quality web systems. Performance, security, and usability. Addison-Wesley; 2002.

[11] Olsina L, Rossi G. Measuring web application quality with WebQEM. IEEE Multimedia 2002:20–45.

[12] Fons J, Pelechano V, Albert M, Pastor O. Development of web applications from web enhanced conceptual schemas. In: Conference on conceptual modeling. LNCS, vol. 2813. Springer-Verlag; 2003. p. 232–45.

[13] Wirsing M, Koch N, Rossi G, Garrido A, Mandel L, Helmerich A, et al. Hyper-UML: specification and modelling of multimedia and hypermedia applications in distributed systems. In: 2nd Workshop on the German–Argentinian bilateral programme for scientific and technological cooperation, Königswinter, Germany, 1999.

[14] Gellersen HW, Gaedke M. Object-oriented web application development. IEEE Internet Comput 1999:60–8.

[15] Gellersen HW, Wicke R, Gaedke M. WebCompostion: an object-oriented support system for the web engineering lifecycle. Comput Networks ISDN Syst 1997;29:1429–37.

[16] Kappel G, Pröll B, Retschitzegger W, Schwinger W. Modelling ubiquitous web applications – the WUML approach. In: International workshop on data semantic in web information systems, Kyoto, Japan, 2001.

[17] Kappel G, Pröll B, Retschitzegger W, Schwinger W. Modelling customizable web applications – a requirement's perspective. In: International workshop on data semantic in web information systems, Kyoto, Japan, 2001.

[18] UML 2.0. www.omg.org.

[19] Garzotto F, Schwabe D, Paolini P. HDM – a model based approach to hypermedia application design. ACM Trans Inform Syst 1993;11(1):1–26.

[20] Chen P. The entity-relationship approach: towards a unified behavior of data. ACM Trans Database Syst 1976;1(1):9–36.

[21] Codd EF. The relational model for database management. Addison-Wesley; 1992.

[22] Baresi L, Garzotto F, Paolini P. Extending UML for modelling web applications. In: Annual Hawaii international conference on system sciences, Maui, USA, 2001. p. 1285–94.

[23] Rossi G. An object oriented method for designing hypermedia applications. PhD Thesis, University of PUC-Rio, Rio de Janeiro, Brazil, 1996.

[24] Isakowitz T, Stohr E, Balasubramanian P. RMM: a methodology for the design of structured hypermedia applications. Commun ACM 1995;38(8):34–44.

[25] Lange D. An object-oriented design approach for developing hypermedia information systems. In: Sprague R, editor. 31st Annual conference on systems science, 1995.

[26] Rumbaugh J. OMT insights: perspectives on modelling from the Journal of Object Oriented Technology. New York: SIGS Books; 1996.

[27] Lange DB. An object-oriented design method for hypermedia information systems. In: 27th Annual Hawaii international conference on system sciences (HICSS'94). IEEE Computer Society Press; 1994. p. 366–75.

[28] Schwabe D, Rossi G. Developing hypermedia application using OOHDM. In: Workshop on hypermedia development processes, methods and models (Hypertext 98), Pittsburgh, USA, 1998.

[29] Schwabe D, de Almenia Pontes R. OOHDM-WEB: rapad prototyping of hypermedia applications. Pontificia Universidade Católica do Rio de Janeiro, MCC 08/98, Rio de Janeiro, Brasil, 1998.

[30] Schwabe D, Rossi G. An object oriented approach to web-based application design. Theory Practice Object Syst 1998;4(4).

[31] Vilain P, Schwabe D, Sieckenius C. Use cases and scenarios in the conceptual design of web application. Technical Report MCC 12/00, Departamento de Informática, PUC-Rio, Rio de Janeiro, Brasil, 2000.

[32] Vilain P, Schwabe D, Sieckenius C. A diagrammatic tool for representing user interaction in UML. In: Lecture Notes in Computer Science, UML'2000, York, England, 2002.

[33] Lima F, Schwabe D. Application modelling for the semantic web. In: LA-WEB 2003 – First Latin American web conference. Santiago (Chile): IEEE-CS Press; 2003.

[34] De Troyer O, Leune C. WSDM: a user-centered design method for web sites. In: Computer networks and ISDN systems, 7th international world wide web conference. Elsevier; 1998. p. 85–94.

[35] De Troyer O, Plessers P, Casteleyn S. Conceptual view integration for audience driven web design. In: WWW2003 conference, Budapest, Hungry, 2003.

[36] De Troyer O, Plessers P, Casteleyn S. Solving semantic conflicts in audience driven web design. In: WWW/Internet 2003 conference, Algarve, Portugal, 2003.

[37] Lee H, Lee C, Yoo C. A scenario-based object-oriented methodology for developing hypermedia information systems. In: Sprague R, editor. 31st Annual conference on systems science. IEEE, 1998. p. 121–38.

[38] Suh W, Lee H. A methodology for building content-oriented hypermedia systems. J Syst Software 2001;56:115–31.

[39] Weidenhaupt K, Pohl K, Jake M, Haumer P. Scenarios in system development: current practice. IEEE Software 1998;2:34–45.

[40] Bieber M, Galnares R, Lu Q. Web engineering and flexible hypermedia. In: 2nd Workshop on adaptative hypertext and hypermedia, 1998.

[41] Olsina L. Building a web-based information system applying the hypermedia flexible process modelling strategy. In: Workshop on hypermedia development processes, methods and models (Hypertext 98), Pittsburgh, USA, 1998.

[42] Conallen J. Building web applications with UML. Addison-Wesley; 1999.

[43] Ceri S, Fraternali P, Bongio. Web modelling language (WebML): a modelling language for designing web sites. Conference WWW9, Mayo. Comput Networks 2000;33(1–6):137–57.

[44] Ceri S, Fraternali P, Bongio A, Brambilla M, Comai S, Matera M. Designing data-intensive web applications. Morgan Kaufman; 2003.

[45] Ceri S, Fraternali P, Matella M. Conceptual modeling of data-intensive web applications. IEEE Internet Comput 2002:20–30.

[46] WebRatio. The CASE tool for the web. Politechnical University of Milano. http://www.webratio.com/sv1.do.

[47] Koch N. Software engineering for adaptive hypermedia applications. PhD Thesis, FAST Reihe Softwaretechnik Vol(12), Uni-Druck Publishing Company, Munich, Germany, 2001.

[48] Kruchten P. The rational unified process. Addison-Wesley; 1998.

[49] ArgoUWE – CASE Tool for modeling web applications. Ludwig-Maximilians-Universität München. http://www.pst.informatik.uni-muenchen.de /projekte/argouwe/. 2004.

[50] UWA requirements elicitation: model, notation, and tool architecture, 2001. www.uwaproject.org.

[51] Cachero C. Una extensión a los métodos OO para el modelado y generación automática de interfaces hipermediales. PhD Thesis, University of Alicante, Alicante, Spain, 2003.

[52] VisualWADE Tool. University of Alicante. http://gplsi.dlsi.ua.es/iwad/ooh_project/index.htm.

[53] Escalona MJ. Modelos y técnicas para la especificación y el análisis de la navegación en sistemas software. PhD Thesis, University of Seville, Seville, Spain, 2004.

[54] Escalona MJ, Mejías M, Torres J, Reina AM. The NDT development process. In: Proceedings of IV international conferences on web engineering. LNCS, vol. 2722. Springer-Verlag; 2003. p. 463–7.

[55] Escalona MJ, Mejías M, Torres J. Developing systems with NDT & NDT-Tool. In: 13th International conference on information systems development: methods and tools, theory and practice, Vilna, Lithuania, 2004. p. 149–59.

[56] Escalona MJ, Mejías M, Torres J, Reina AM. NDT-Tool: a tool case to deal with requirements in web information systems. In: Proceedings of IV International Conferences on Web Engineering. LNCS, vol. 2722. Springer-Verlag; 2003. p. 212–3.

[57] Escalona MJ, León A, Martín A, Mejías M, Torres J, El Tesauro de Patrimonio Histórico de Andalucía. In: IV Jornadas de Bibliotecas Digitales, Alicante, España, 2003. p. 105–14.

[58] Villadiego D, Escalona MJ, Torres J, Mejías M. Aplicación de NDT al sistema para el reconocimiento, declaración y calificación del grado de minusvalía. Internal Report LSI-2004-02, University of Seville, 2004.

[59] Mecca G, Atzeni P, Crescenzi V. The ARANEUS guide to web-site development. Technical Report, Universidad de Roma, 03 1999, Roma, Italia, 1999.

[60] Nanard J, Nanard J. Hypertext design environments and the hypertext design process. Commun ACM 1995;38(8):49–56.

[61] Thomson J, Greer J, Cooke J. Algorithmically detectable design patterns for hypermedia collections. In: Workshop on hypermedia development process, methods and models. Hypermedia, 1998.

[62] Liddle SW, Embley DW, Woodfiel SN. A seamless model for object-oriented systems development. In: First international workshop on web-oriented software technology, Valencia, 2001.

[63] Liddle SW, Embley DW, Woodfiel SN. An active, object-oriented, model-equivalent programming language. In: First international workshop on web-oriented software technology, Valencia, 2001.

[64] Eklund J, Lowe D. Using partial design to elicit requirements in web development – a survey of commercial practice, 2001.

[65] Lowe D, Eklund J. Client needs and the design process in web projects. In: Web engineering track of the WWW2002 conference, 2002.

[66] Garzotto F, Mainetti L, Paolini P. Hypermedia design analysis, and evaluation issues. Commun ACM 1995;38(8):74–86.

[67] Escalona MJ, Koch N. Requirements engineering for web applications: a comparative study. J Web Eng 2004;2(3):193–212.

[68] Cachero C, Koch N. Navigation analysis vs. navigation design. An example for discussion. Internal Report, University of Alicante, TR-Ap02b, Alicante, Spain, 2002.

[69] Cachero C, Koch N. Conceptual navigation analysis: a device and platform independent navigation specification. In: 2nd International workshop on web-oriented software technology, Málaga, Spain, 2002.

[70] Jacobson I, Booch G, Rumbaugh J. The unified software development process. Addison-Wesley; 1999.